CSE 573: Final Project Report
Amazon Fake Review Detection

Abhay Manojkumar Jayani
SCAI
Arizona State University
Tempe, USA
ajayani1@asu.edu

Darshil Shaileshkumar Shah
SCAI
Arizona State University
Tempe, USA
dshah40@asu.edu

Emma Hanretty
SCAI
Arizona State University
Tempe, USA
ehanrett@asu.edu

Fenny Zalavadia
SCAI
Arizona State University
Tempe, USA
fzalavad@asu.edu

Pratik Giri
SCAI
Arizona State University
Tempe, USA
pgiri4@asu.edu

Venkata Pavan Kalyan Gadekari
SCAI
Arizona State University
Tempe, USA
gpavanka@asu.edu

***Abstract*—As e-commerce has become more and more popular, an increasing number of people utilize websites like Amazon and Yelp to inform their purchasing decisions. These sites use rating and recommendation systems to aid users in decision-making. Due to this, malicious actors began writing spam reviews to influence user opinion on a product, which could provide a significant monetary benefit to the product owner. Initially, opinion spam–or spam reviews that are obviously fake–was the most popular method of spamming. However, more recently deceptive opinion spam–users writing fake reviews that sound genuine–has become prominent and is much harder to detect. Thus, it is vital to develop methods to distinguish real reviews from fake ones. This project aims to propose models that help filter spam reviews from real ones.**

***Keywords*—fake review detection, supervised classification, deceptive opinion spam, naive bayes, random forest, support vector machines, artificial neural networks, CNN, LSTM**

I. INTRODUCTION

The influence and reach of eCommerce sites have been expanding as more and more people have obtained access to the internet. These websites commonly allow people to write reviews of products that they have purchased, which helps the manufacturers produce better products and inform others' purchasing decisions. However, as these sites have become more popular, the impact of fake reviews has increased and led to a need for research into these fields.

Deceptive opinion spam refers to fake reviews that have been written with the purpose of sounding genuine, and are therefore harder to detect than other kinds of fake reviews like spam. Websites have developed methods of detecting this opinion spam to help minimize the influence of such spam on users' purchasing choices, and is especially important in eCommerce fields like restaurants and online shopping.

Machine learning has provided numerous techniques to detect opinion spam. These include supervised machine learning, which connects features to the reviewers and their reviews to identify fictitious ones. Unsupervised machine learning focuses more on identifying features of the reviews themselves in an attempt to detect opinion spam, but they are usually not as effective at detecting deceptive opinion spam.

In this paper, we will explore numerous approaches to detecting fake reviews - including Naive-Bayes, Random Forest Generator, Support Vector Machines, Neural Networks, CNN, and LSTM - and compare their performance and accuracy. We utilized a large, publicly available dataset from Amazon.com to perform these algorithms and analyses.

II. PROBLEM STATEMENT

If you want to buy a product nowadays, you will almost certainly read product reviews first. If the majority of the evaluations are positive, he or she is quite likely to purchase it. However, if the majority of the evaluations are bad, he or she will almost certainly go for a different product. Unfortunately, this provides great incentives for imposters to take advantage of the system by publishing fake reviews in order to promote or denigrate specific items and services. During the course of this project, we aim to study various techniques that the system currently utilizes to detect fake reviews. We also intend to conduct feature engineering to determine the most important features that aid in the detection of fake reviews. We want to test several machine learning and deep learning algorithms on the Amazon Dataset and report on their accuracy. Finally, we intend to compare different existing algorithms with our current algorithm with respect to accuracy and report the findings.

III. RELATED WORKS

In the literature review of Fake Review Detection, we looked for the current state of the art. From the literature review of papers on Fake Review Detection, we found a pattern followed by most of the papers. The procedure starts by performing basic steps of Exploratory Data Analysis, followed by pre-processing steps. Pre-processing steps include checking null values, text processing (such as removing stop words, eliminating noise, such as HTML Tags, abbreviations, punctuations, numbers from the data, capitalization, performing Stemming or Lemmatization, Performing feature engineering, handling unbalanced data for some cases). Followed by classification methods.

Previous work done in the field mainly focuses on supervised learning using word unigrams and bigrams, also known as linguistics features and features of user behavior.

User behavior includes features, such as standard deviation of ratings, frequency of words in reviews, etc.

Wang, Z et. al [1] in their paper “Fake Review Detection Based on Multiple Feature Fusion and Rolling Collaborative Training” starts by extracting various user-centric features from their dataset with the aim of increasing training accuracy. Instead of using linguistics features such as unigrams and bigrams, they used Latent Dirichlet Allocation (LDA). They analyzed the impact of each feature for a different set of classifiers. For classification, they used logistic regression, linear discriminant analysis, multinomial Naïve Bayes, support vector machine, and neural networks. For their research, they trained seven classifiers and compared those classifiers based on testing accuracy. Among the used classifiers, neural networks performed the best with a detection accuracy rate of 81.92%.

J. Fontanarava et. Al [2] was recently presented at the international conference of Data Science and Advanced Analytics. The paper compared older approaches for the feature selection along with a newer approach of selecting review–and review-centric–features. The paper tested the results of those features using the Random Forest Classifier and provided the analysis. The main purpose behind using the Random Forest Classifier was to compare the results by considering both well-known features along with these new sets of features on supervised and unsupervised classification. The results showed the effectiveness of new features by using the Random Forest Classifier to detect singleton fake reviews.

IV. DATA

In this project, we are using a dataset generated in "Creating and detecting fake reviews of online products. Journal of Retailing and Consumer Services (2022)"[3]. Two language models are used to generate the fake reviews, ULMFit and GPT-2 and are compared to each other using different metrics. GPT-2 outperformed ULMFit in all the relevant metrics and hence GPT-2 is used to generate the fake reviews. The complete dataset includes 40,432 reviews for top 10 categories from Amazon. These categories include Home and Kitchen, Electronics, Books, etc. Out of the total, 20,216 are fake reviews and 20,216 are real i.e., original reviews. Since there are an equal number of original and fake reviews, this is a well balanced dataset. Each datapoint has four fields: *Category*, *rating*, *label* and *text*. *label* column holds if the review is fake or original by maintaining the column values to CG and OR respectively. We will use these column information for extracting useful features for our model training.

TABLE I
DATA PROFILE OF AMAZON DATASET

	No of reviews	% of total
Genuine	20216	50.00%
Fake	20216	50.00%
Total	40432	100.00%

V. SYSTEM ARCHITECTURE

In this project, we are using the below shown procedure to visualize the problem. The first step is the data pre-processing of the Amazon Dataset. We have performed various pre-processing techniques such as stop words removal, noise elimination, stemming, and lemmatization to remove unnecessary data. The second step is feature extraction and selection to extract important features from the data. For this step, we have implemented various word embedding techniques such as Bag of Words (BoW), TF-ID, Word2Vec, and BERT. After extracting the features, We have implemented supervised and unsupervised classification algorithms such as Support Vector Machine, Naive Bayes, Random Forest, Neural networks, and CNN. And analyzed the model’s performance using the extracted features. The last step is the evaluation of the models. For that, we have used various performance metrics such as precision, recall, F-measure, true positive rate, confusion matrix, etc.

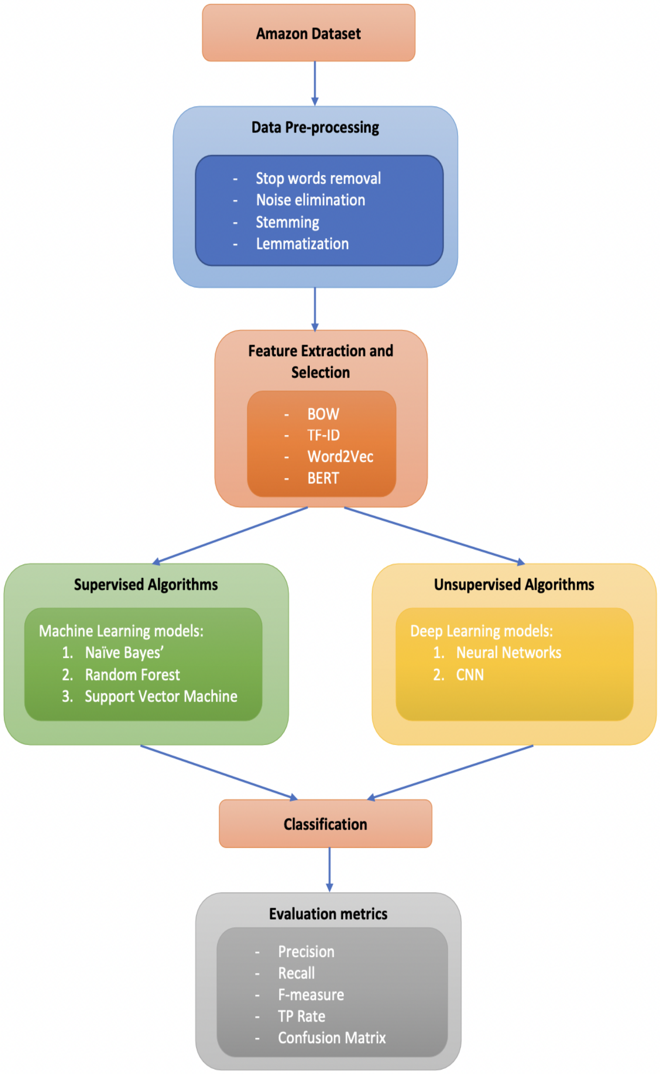


Fig. 1. Solution and System Architecture

VI. DATA PRE-PROCESSING AND FEATURE ENGINEERING

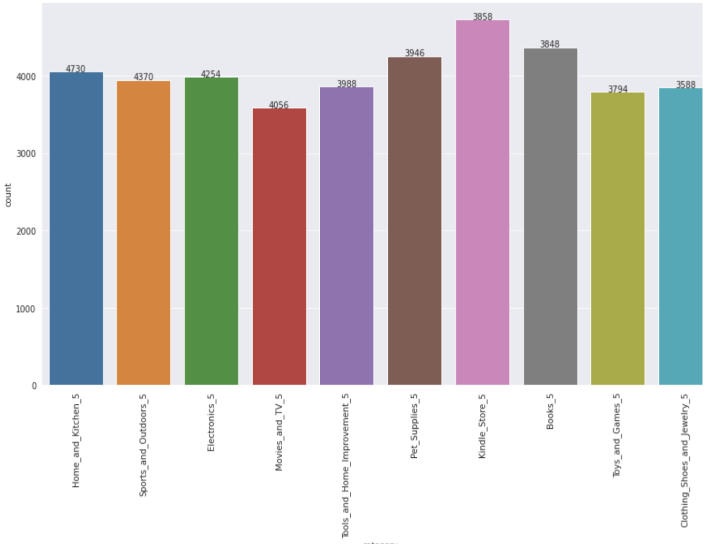


Fig. 2. Data visualization for users, products and date of review.

A. Data Pre-processing

During the literature review we have seen some common pre-processing steps performed. Our data contains 4 features, Category, rating, label, and text. The dataset is well balanced based on the class label as well as category values. It contains 20,216 real reviews and the same number of computer generated reviews. As shown in Fig. 2 the distribution of the data based on the class Category is also balanced. Along with that the dataset does not contain any missing or erroneous values. So we did not perform any pre-processing steps for handling the imbalance data and missing values.

We mainly performed various pre-processing steps on the text column, which has the original review text. After analyzing the text data we find out that the text contains different types of noise, non-useful words, as well as the same word in different forms.

To start with, we first converted the text reviews into lowercase letters. After converting those reviews into lowercase letters we used the Stop words list provided by NLTK library. From that we eliminated a few words which can change the meaning of the entire review.

Subsequently, We thoroughly analyzed the data to identify different types of noise. The reviews contain unnecessary HTML tags, punctuation marks, and digits. We created regular expressions to eliminate this type of noise from the reviews. Sometimes removing the punctuation mark '-' can change the entire meaning of the sentence or word. So we took care of that while creating the regular expressions.

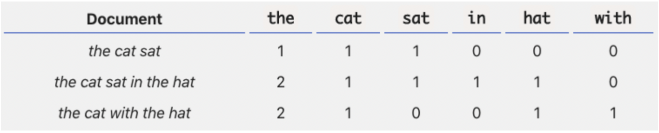
Lastly, we used Lemmatization and Stemming techniques to extract the base form of the word by removing affix from that. We have used WordNet Lemmatizer and Porter Stemmer provided by the NLTK library to convert the words into their root form.

We further performed feature engineering and extraction tasks which are explained in the following section.

B. Feature Extraction

After data preprocessing, we did feature engineering by selecting important features. For selecting and extracting important features, we have used the following techniques:

1. **Bag of Words (BOW):** The bag-of-words (BOW) model is a representation that turns arbitrary text into fixed-length vectors by counting how many times each word appears. This process is often referred to as vectorization.



Document	the	cat	sat	in	hat	with
the cat sat	1	1	1	0	0	0
the cat sat in the hat	2	1	1	1	1	0
the cat with the hat	2	1	0	0	1	1

Fig. 3. Bag of Words sample output

2. **Term Frequency-Inverse Document Frequency (TF-IDF):** is a statistical measure that evaluates how relevant a word is to a document in a collection of documents. This is done by multiplying two metrics: how many times a word appears in a document, and the inverse document frequency of the word across a set of documents.TF-IDF for a word in a document is calculated by multiplying two different metrics:
   a. The **term frequency** of a word in a document.
   b. The **inverse document frequency** of the word across a set of documents. This means, how common or rare a word is in the entire document set.
3. **Word2Vec:** It measures text similarity using cosine similarity techniques and word clustering.A Word2Vec model learns meaningful relations and encodes the relatedness into vector similarity.
4. **tSNE** (t-distributed stochastic neighbor embedding**):** TSNE is pretty useful when it comes to visualizing similarity between objects. It works by taking a group of high-dimensional (100 dimensions via Word2Vec) vocabulary word feature vectors, then compresses them down to 2-dimensional x,y coordinate pairs. The idea is to keep similar words close together on the plane, while maximizing the distance between dissimilar words.

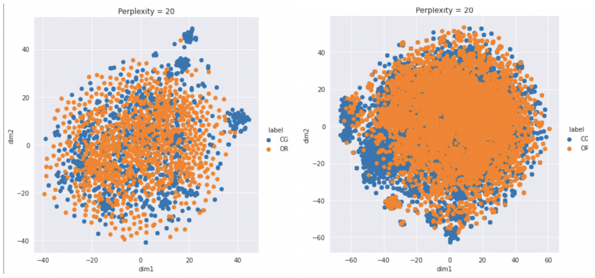


Fig. 4. tSNE dimension reduction using BOW representation

We have used the Dimensionality reduction technique called tSNE (t-Distributed Stochastic Neighbor Embedding) for visualizing high dimensional data. So, here we have tried to visualize 2000 data points after BOW with perplexity 20. Perplexity is a tunable parameter, it gives a guess about the number of close neighbors each point has. Similarly , we tried with 10,000 data points also, and we can see it's highly dense.

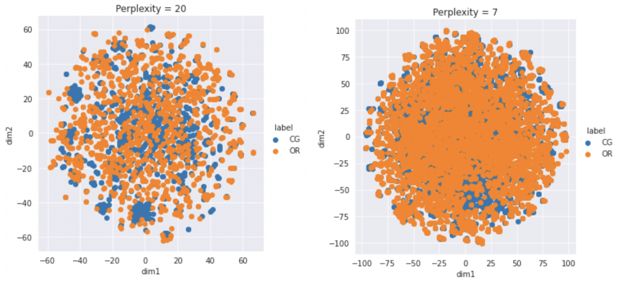


Fig. 5. tSNE dimension reduction using TF-IDF representation

Something similar we did for TF IDF data as well. In Fig. 5, on the left we can see data is scattered and loosely coupled, whereas on the right it's densely connected.

VII. MODELS, TRAINING & EVALUATION

1) *Models and Training:*

A. *Machine Learning Models*

a. SVM:

This model is a supervised learning model with associated learning algorithms. SVM takes selected features as input and tries to find the best hyperplane which separates Computer generated and original reviews with maximum margin distance. While implementing the algorithm, we selected the kernel to be linear. We found out the accuracy of this model to be 87.12%, which is the highest among the Machine learning models we have implemented. The accuracy and other evaluation metrics obtained by this method is depicted in Section VIII.

b. Naive Bayes:

This is a Classification technique based on Bayes theorem with an assumption of independence among predictors. This model is particularly useful for very large datasets. We will be providing input in terms of selected features to the naive bayes model where the model predicts the probability of a datapoint belonging to Original or Computer Generated review. We found out the accuracy of this model to be 85.05%. The accuracy and other evaluation metrics obtained by this method is depicted in Section VIII.

c. Random Forest:

This is an ensemble learning method for classfication. Input to this model is the selected features wherein a tree is built using edges as criteria to belong to Original review or Computer generated review and nodes represent the important features in specific order. We found out the accuracy of this model to be 85.64%. The accuracy and other evaluation metrics obtained by this method is depicted in Section VIII.

B. *Deep Learning Models*

a. Neural Network:

For initial training we used an Embedding Layer with dimension of 50, mapping each word to a vector representation. We decided to use a fully-connected Dense Layer with 2 output classes. But in order to plug the Embedding layer to the Dense Layer, we need to compute the vector representation of all the words in each sequence and average them before feeding them forward. For the Dense Layer, Sigmoid activation function is used because of its performance when the output layer has two classes (binary classification). For performance measurement of the deep learning models we decided to use ROC-Area Under the Curve (AUC) as it works better for binary classification problems. Section V depicts the performance of NN.

b. LSTM:

RNN has a very important advantage of backward propagation. LSTM [4] is a special case implementation of RNN eliminating the problem of vanishing gradient [5]. We have implemented LSTM with its hidden layer having 100 outputs. As before this layer is connected to a Dense layer having 2 outputs. It is important to note that we were able to improve the performance of this model significantly by varying the hyperparameters like number of epochs, batch size, etc yielding the highest accuracy among the rest. The accuracy obtained is depicted in Section VIII.

c. CNN + LSTM:

This architecture involves using Convolutional Neural Network (CNN) layers for feature extraction on input data combined with LSTMs to support sequence prediction (see Fig. 6) [6]. Text classification using this method was studied in [7]. 1D CNN is designed to effectively extract the features of the text sequence data. In addition, the extracted features are then processed by the LSTM layers in order to further extract the temporal features. Finally, the output features are fed into a Dense layer. In our implementation, both convolution layer and LSTM layer have 64 output channels connecting them to a Dense Layer having 2 outputs. The accuracy obtained by this method is depicted in Section VIII. It should be noted that the accuracy can be improved by working with a much larger dataset.

d. BERT:

BERT stands for Bidirectional Encoder Representations from Transformers. To implement this model, we have added special tokens such as "CLS" and "SEP" to understand the input sentences. "CLS" denotes the start of sentences, whereas "SEP" denotes the beginning of the next sentence. These tokens were added to separate the individual sentences from each other. After that, we have tokenized the data using BERT tokenizer to convert the data into index numbers in BERT vocabulary. After training the BERT model and evaluating the same on the test data, we achieved the 90.50 % accuracy.

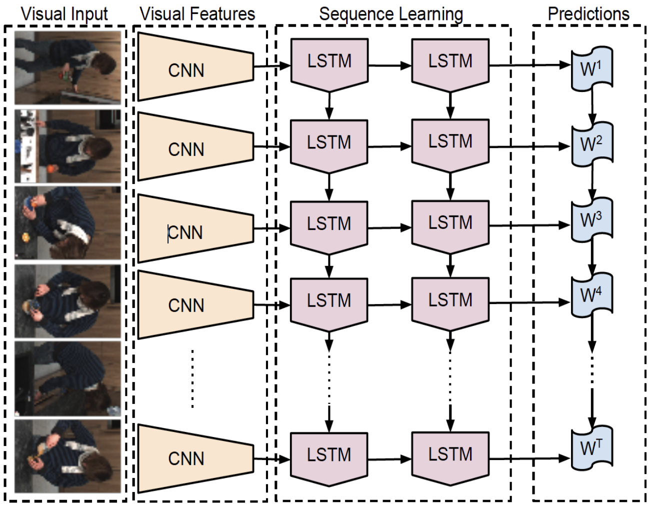


Fig. 6. CNN LSTM architecture formerly referred to as Long-term Recurrent Convolutional Networks (LRCNs)

2) *Evaluation Metrics*

The standard of a domain system can be evaluated by comparing predictions obtained from a test set of known reviews. These systems are typically measured using performance metrics such as precision, recall, F-measure, and many others.

- **Precision:** It is the measure of exactness, which determines the fraction of relevant items retrieved out of all items. Precision (P) is the proportion of legitimate reviews that are truly original.
- **Recall:** It is a measure of completeness, which determines the fraction of relevant items retrieved out of all relevant items. Recall (R)is the proportion of all legitimate reviews.
- **F-measure:** It is defined as the HM (harmonic mean) of recall (R) and precision (P).
- **True Positive Rate:** It is the percentage of fake reviews from the total dataset that are correctly classified.
- **Confusion Matrix:** This is the matrix that summarizes the predictions into 4 classes:
  True Positive, True Negative, False Positive, and False Negative.

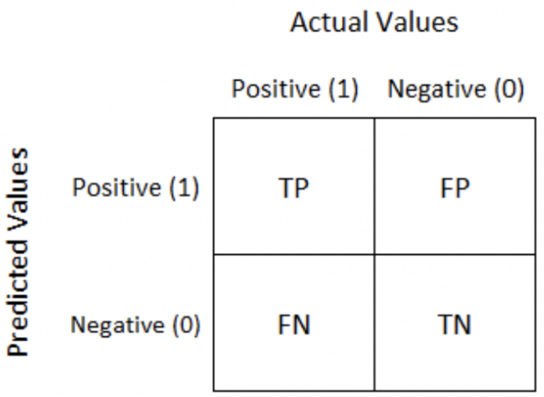


Fig. 7. Confusion Matrix

VIII. COMPARATIVE ANALYSIS

This section focuses on comparing results obtained by the algorithms we ran on the data set. Overall, the Naive-Bayes model had the lowest overall accuracy of about 85%, while the LSTM model had the highest accuracy at almost 98%. We also compared the average accuracies of machine learning models, including Naive-Bayes, SVM, and Random Forest Classifier, versus those of deep learning algorithms, including LSTM, CNN+LSTM, and neural networks. It was found that deep learning algorithms had a much higher overall accuracy, at almost 96%, than the machine learning algorithms which averaged around 87% accuracy.

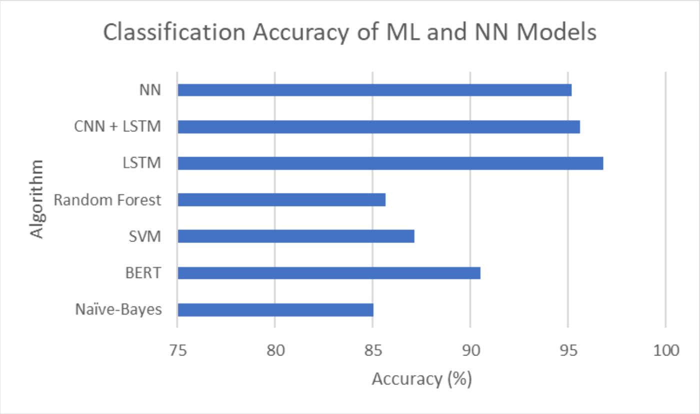


Fig. 8 Comparison between Accuracies of different models

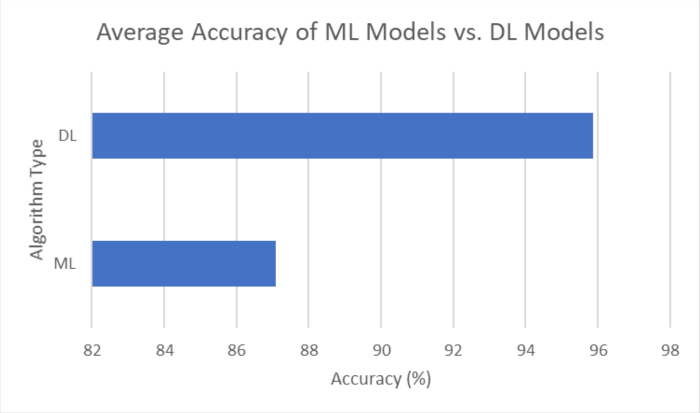


Fig. 9. Comparison between Accuracies of ML and DL Models

Our results support this initial analysis.

TABLE II

AVERAGE RESULTS FOR THE NAIVE BAYES MODEL

Naive Bayes	Precision	Recall	F1 score	Accuracy
Real (0)	0.89	0.79	0.84	85.05
Fake (1)	0.82	0.91	0.86	

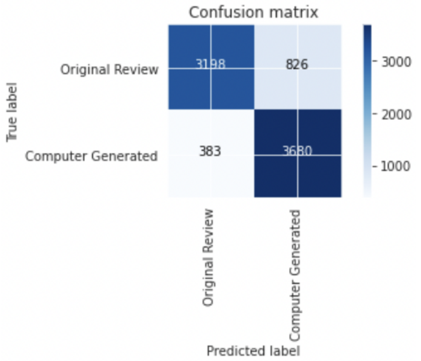


Fig. 10. Confusion Matrix of Naive Bayes

TABLE III

AVERAGE RESULTS FOR THE RANDOM FOREST MODEL

Random Forest	Precision	Recall	F1 score	Accuracy
Real (0)	0.87	0.83	0.85	85.64
Fake (1)	0.84	0.88	0.86	

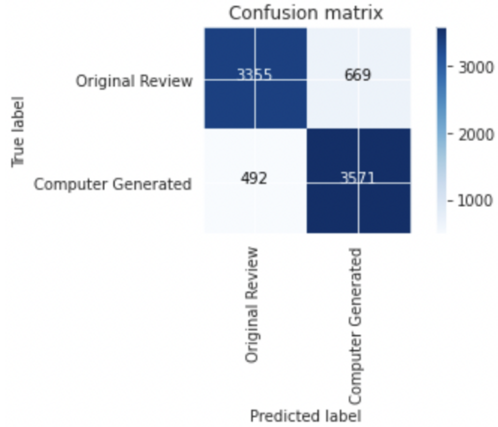


Fig. 11. Confusion Matrix of Random Forest Classifiers

TABLE IV
AVERAGE RESULTS FOR THE SUPPORT VECTOR MACHINE

SVM	Precision	Recall	F1 score	Accuracy
Real (0)	0.87	0.87	0.87	87.12
Fake (1)	0.87	0.87	0.87	

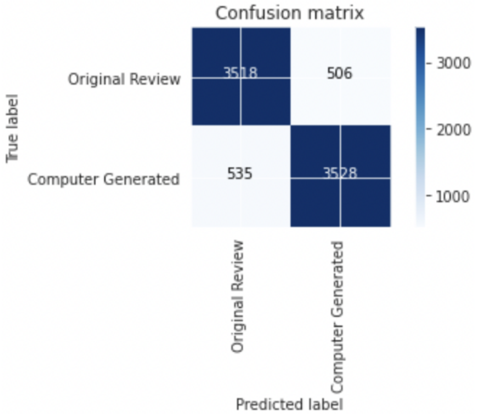


Fig. 12. Confusion Matrix of SVM

IX. CHALLENGES

To make the text review data noise-free, we decided to filter this data using a Regular Expression. But coming up with the final RegEx took several iterations. To come up with the final used RegEx, for a sample of data, we manually checked the output after applying the expression. This was done to verify not only to detect if there is some more noise that we can remove but also to verify if the output is not semantically different from the original.

An important challenge that we faced was about the absence of information in the used dataset. The dataset that we have used, though being a very well balanced one, does not have any user information. This is the case because the authors of the dataset have generated it with the aim of using it for text classification. So the absence of this user information metadata, had to be compensated by looking at the Amazon Fake Review detection problem as a purely text classification problem. Consequently, all the stages in data preprocessing and applying different ML and DL models were modeled and executed keeping this approach in mind.

Another dataset challenge that we faced was regarding the size of the dataset. Dataset used has 40,432 data points. Typically, for a text classification, we have seen in the literature review that the dataset size is in the range of 100,000. This size is generally required by the Neural Networks especially in Deep Learning and tree based algorithms to train the model adequately.

One major problem we faced during the implementation of the Deep Learning models and lack of resources and abundant amount of time required to train and test those models. It was impossible to run those models on our personal computers. We used Agave clusters [8] provided by the university to run our models.

X. INDIVIDUAL CONTRIBUTION

This project and completing all identified tasks has been a team effort, where every member has been a valuable asset to the team. Table V summarizes the task division and the members responsible for it.

TABLE V
DIVISION OF WORK

Task	**Ownership**
Literature Review	Abhay, Darshil, Emma, Fenny, Pratik, Pavan
Data Analysis and Exploration	Abhay, Darshil, Emma
Data Cleaning and Processing	Pratik, Abhay, Darshil
Feature Engineering	Fenny, Abhay, Darshil
Model Creation and Implementation	Fenny, Pratik, Pavan
Evaluation	Pavan, Fenny
Comparative Analysis of Models	Pavan, Pratik, Emma
Future Scope	Fenny
Presentation	Abhay, Darshil, Emma, Fenny, Pratik, Pavan
Final Report	Abhay, Darshil, Emma, Fenny, Pratik, Pavan

The individual contributions for each team member has been listed below.

- **Pratik Giri** was involved in Literature Review by exploring past related works on Fake Review Detection as well as examination of the Amazon Dataset in the initial phase of the project. In the implementation phase, he contributed towards creation, implementation and improvement of the Neural Network model, LSTM model and CNN+LSTM model. Finally he took part in the project presentation and maintenance of the report.
- **Emma Hanretty** has worked on literature review. She identified popularly used algorithms and features in the field of fake review detection and how these data were graphed. Next she worked on comparing the accuracies of the different algorithms implemented and graphing these comparisons. Finally she participated in the project presentation and final report.
- **Fenny Zalavadia** has been involved mainly in the feature extraction as an individual task. She worked on extracting the features from the Amazon dataset. Also, she was involved in applying different feature engineering techniques like BOW, TF-IDF and tSNE on pre-processed data. Apart from this, she was involved in the project demo presentation and docu- mentation.

- **Venkata Pavan Kalyan Gadekari** has worked on machine learning model training and implementation such as Support Vector Machines(SVM), Naive Bayes Classifier, and Random Forest Classifier. In the next phase, he has been involved in plotting Classification reports, Confusion matrix, extracting evaluation metrics and comparative analysis of these ML models. Apart from this, he has also been involved in the Project presentation and documentation.
- **Darshil Shah** has done literature reviews and summarizes the previous methods/techniques used in the field to identify the fake reviews. He performed exploratory data analysis on Amazon's fake review dataset. After thoroughly analyzing the data identified data cleaning and preprocessing tasks to make data more suitable for machine learning models. Apart from this he was involved actively in the creation and presentation of the project presentation and report.
- **Abhay Jayani** has worked on designing the overall system architecture of the project. Also, he has worked on how to apply the BERT model on the dataset and worked on implementation of the BERT model from data preparation for the BERT model to training and testing of the BERT model. Apart from this, he was also involved in the project presentation, demo presentation, and the final project report.

XI. CONCLUSION

In this project we have performed fake review detection using supervised classification models and Deep Learning models. We used the Amazon dataset and analyzed the data to extract features from it. Extracted features are a bundle of different types of features;Finally we compared results from seven different models; Naive Bayes, Support Vector Machine, Random Forest Classifier, Neural Networks, LSTM, CNN + LSTM, BERT. The final results show that LSTM and CNN + LSTM classifiers performed better than the other models.. Our solution used machine learning libraries in Python such as keras (for artificial neural network training) and scikit-learn’s Stratified K-fold Cross validation (for naive bayes, random forest and decision trees) to ensure the ratio of the two classes are same in the test and train data. Additionally we also use numpy and pandas during the feature extraction and feature analysis phases.

ACKNOWLEDGMENT

We would like to thank our professor Hasan Davulcu for his continued guidance and support throughout the course of this project.

REFERENCES

[1] Wang, Jingdong, et al. "Fake review detection based on multiple feature fusion and rolling collaborative training." IEEE Access 8 (2020): 182625-182639.

[2] Fontanarava, Julien, Gabriella Pasi, and Marco Viviani. "Feature analysis for fake review detection through supervised classification." 2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA). IEEE, 2017.

[3] Joni Salminen, Chandrashekhar Kandpal, Ahmed Mohamed Kamel, Soon-gyo Jung, Bernard J. Jansen, Creating and detecting fake reviews of online products, Journal of Retailing and Consumer Services, Volume 64, 2022, 102771, ISSN 0969-6989, https://doi.org/10.1016/j.jretconser.2021.102771

[4] Hochreiter, S., Schmidhuber, J., “Long Short-Term Memory”, Neural Computation 9 (8), 1997, pp. 1735–1780

[5] Gers, F. A., Schraudolph, N. N., Schmidhuber, J., “Learning Precise Timing with LSTM Recurrent Networks”, Journal of Machine Learning Research 3, 2002, pp. 115–143

[6] Donahue, Jeffrey, et al. "Long-term recurrent convolutional networks for visual recognition and description." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.

[7] Y. Luan and S. Lin, "Research on Text Classification Based on CNN and LSTM," 2019 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), 2019, pp. 352-355, doi: 10.1109/ICAICA.2019.8873454.

[8] Large-scale Computational Resources, ASU, https://cores.research.asu.edu/research-computing/about